UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/711,816	10/07/2004	Ching-Hai Hung	IFTP0005USA	5815

27765 7590 12/16/2010
NORTH AMERICA INTELLECTUAL PROPERTY CORPORATION
P.O. BOX 506
MERRIFIELD, VA 22116

EXAMINER	
BIRKHIMER, CHRISTOPHER D	

ART UNIT	PAPER NUMBER
2186	

NOTIFICATION DATE	DELIVERY MODE
12/16/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

Patent.admin.uspto.Rcv@naipo.com
mis.ap.uspto@naipo.com

Office Action Summary	Application No. 10/711,816	Applicant(s) HUNG, CHING-HAI	
	Examiner CHRISTOPHER D. BIRKHIMER	Art Unit 2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 June 2010*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

The current Office Action is in response to the Request for Continued Examination submitted 12/06/2010. The Examiner acknowledges the amendments to claim 1. Claims 1 – 24 are currently pending in the case

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

3. Claims **1, 3, 7, 11 – 14, 19 – 21, and 23- 24** are rejected under 35 U.S.C. 103(a) as being unpatentable over DeKoning et al (Pat 6,467,023) in view of Randall et al. (Pat 6,530,031).

With regard to **claim 1**, DeKoning teaches a method for redundant array of independent disks (RAID) **[100, Fig 1; Column 5, Lines 19 - 21]** consistency initialization **[Fig 3 and Fig 4]** comprises:

creating a RAID **[100, Fig 1; Column 5, Lines 19 - 21]**, including setting a RAID configuration of the RAID **[“configuration information”, Column 3, Lines 34 – 41]** and storing progress states of the initialization of the RAID **[300, 302, 304, 306, 307, Fig 3, This describes the function of storing the state of the initialization of the LUN for the RAID system]**;

a plurality of initialization regions of the RAID **[100, Fig 1; Column 5, Lines 19 – 21; Fig 3, The portions of the LUN are the initialization regions of the RAID since the portions are initialized]** in which after the initialization is started and before the consistency initialization **[Fig 3 and Fig 4]** is completed, the RAID **[100, Fig 1; Column 5, Lines 19 – 21]** is allowed to be accessed while the consistency initialization is in progress **[Column 3, Lines 41 – 43]** and data in the initialization regions of the RAID **[100, Fig 1; Column 5, Lines 19 – 21; Fig 3, The portions of the LUN are the initialization regions of the RAID since the portions are initialized]** are made consistent with one another by the consistency initialization **[Fig 3 and Fig 4; Column 7, Lines 25 – 41, This shows what is involved in step 210 of the initialization process. In a mirroring RAID setup the different regions are made identical with the method of mirroring data from a source to a backup location. In another initialization process the regions are made identical by setting all the data and backup data location to zeros. Both initialization processes provide for initialization where initialization regions are made consistent with one another. The one uses replication of data from one location to another and the other method sets the regions that are initialized to a default value]**.

However, DeKoning does not specifically disclose the limitation of creating an initialization progress table for storing progress states of the initialization, the initialization progress table including a plurality of fields for storing initialization states of each of a plurality of initialization regions of the RAID so as to indicate which initialization regions have been initialized by regional initialization and which initialization regions have not yet been initialized, and after the initialization progress table is created and before the consistency initialization is completed the RAID is allowed to be accessed while the consistency initialization is in progress.

Randall discloses creating an initialization progress table **[500, Fig 5]** for storing progress states **[502, 504, and 506, Fig 5, Each item in the table is a progress indicator]**; wherein the initialization progress table **[500, Fig 5]** includes a plurality of fields **[502, 504, and 506, Fig 5]**, for storing initialization states of each of a plurality of initialization regions of the RAID so as to indicate which initialization regions have been initialized by regional initialization and which initialization regions have not yet been initialized **[500, Fig 5; Column 4, Lines 55 – 58; Column 5, Lines 5 – 16, The figures and text shows the table contains a plurality of entries for regions that are initialized and regions that are not initialized. Entries that have a duration time are initialized and entries that don't have a duration time are not initialized]**, and after the initialization progress table is created **[500, Fig 5]** and before the consistency initialization is completed the RAID is allowed to be accessed while the consistency initialization is in progress **[DeKoning, Column 3, Lines 41 – 43; Randall, Fig 5; This**

teaches the table is created when initialization is started and that the storage system can also be accessed while the system is still in an initialization mode].

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of Randall in DeKoning, because it allows the system to maintain multiple indications of initialization and allows the system to resume from multiple initialization steps and at the progress level the initialization was at when it was halted and the table of Randall provides data that allows the user of the system to determine if there are problems with the system automatically **[Column 2, Lines 19 - 29]**.

With regard to **claim 3**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** comprises an induced consistency initialization which comprises steps of:

detecting, when the RAID receives an I/O **["PROCESS I/O REQUEST", Fig 4]**, whether the initialization region(s) that is(are) associated with the I/O has(have) completed with the regional initialization **[405, Fig 4]**;

initializing the initialization region(s) that is(are) associated with the I/O first if the initialization region(s) that is(are) associated with the I/O has(have) not completed with the regional initialization **[408, Fig 4; Column 9, Lines 9 – 18 and 35 – 51, This shows the next step performed from 405 when the initialization for the given region is not completed is that the initialization is completed for the given region to generate new redundancy information]**.

With regard to **claim 7**, DeKoning teaches the step of performing a consecutive consistency initialization **[310, Fig 3; Fig 4, This shows the initialization is**

consecutive until it is complete] on the initialization regions that have not yet been completed with the regional initialization **[Fig 3, The figure shows that initialization starts at the beginning of the LUN and ends once it has initialized the whole LUN]**.

With regard to **claim 11**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** further comprises a consecutive consistency initialization **[Fig 3, he figure shows that initialization starts at the beginning of the LUN and ends once it has initialized the whole LUN]** and after the initialization progress is created, the consecutive consistency initialization **[Fig 3, The figure shows that initialization starts at the beginning of the LUN and ends once it had initialized the whole LUN]** is allowed to be start anytime **["CHECKPOINT RESTART", and 316, Fig 3; Column 8, Lines 30 – 38, This shows that once the initialization progress is created the consecutive initialization is allowed to start]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

With regard to **claim 12**, DeKoning teaches the RAID **[100, Fig 1; Column 5, Lines 19 - 21]** is allowed I/O accessing **[Column 3, Lines 50 – 53]** before the consecutive consistency initialization **[Fig 3]**.

With regard to **claim 13**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** comprises dividing a data space of member disks **[108, Fig 1]** into a plurality of initialization regions **["PORTION", 300 and 310, Fig 3]** and performing the regional initialization **["PORTION", 300, 302, 304, 306, 307, 308, and 310, Fig 3]** on the initialization regions **["PORTION", 300, 302, 304, 306, 307, 308, and 310, Fig 3]**.

With regard to **claim 14**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** comprises dividing a data space of member disks **[108, Fig 1]** into a plurality of initialization regions **[“PORTION”, 300 and 310, Fig 3]** and performing the regional initialization on the initialization regions **[“PORTION”, 300, 302, 304, 306, 307, 308, and 310, Fig 3]**.

With regard to **claim 19**, DeKoning teaches if a member disk has failed and a new member disk **[Column 9, Lines 60 – 62]** is used to perform a rebuilding of the RAID **[100, Fig 1; Column 5, Lines 19 - 21]** before the completion of the consistency initialization **[Column 9, Lines 55 – 58]**, the rebuilding only has to perform on the regions which have been completed with the consistency initialization and the rebuilding on the regions which have not been completed with the consistency initialization can be performed by the consistency initialization **[308 and 312, Fig 3; 502 and 504, Fig 5; Column 10, Lines 1 – 11, This shows the rebuilding is done on the redundancy information affected by the replace which would only include initialized regions and then the LUN is marked as fully initialized once initialization is complete]**.

With regard to **claim 20**, DeKoning teaches when an I/O operation **["PROCESS I/O REQUEST", Fig 4]** accesses the RAID **[100, Fig 1; Column 5, Lines 19 - 21]** is a read operation **[The I/O request indicates both read and write requests]**, and the initialization region of the RAID **[100, Fig 1; Column 5, Lines 19 - 21]** to be accessed by the I/O has not been initialized yet **[405 and 407, Fig 4]**, no consistency initialization is performed on the initialization region **[407, Fig 4]**, and a value of zero or a predetermined value will be returned directly **[407, Fig 3, The I/O request is a**

predetermined value since it is the same I/O request at the beginning of the process and not a newly determined I/O request. The predetermined I/O request is returned directly to a queue to be processed later].

With regard to **claim 21**, DeKoning teaches a RAID **[100, Fig 1; Column 5, Lines 19 - 21]** performs an I/O operation **["PROCESS I/O REQUEST", Fig 4]** and causes an induced consistency initialization **[408, Fig 4; Column 9, Lines 11 – 18]**, if the induced consistency initialization has been completed but the I/O operation has not been completed while the initialization progress has been updated and written into member disks of the RAID, the updated initialization progress will not be written into the member disks again due to completion of the I/O operation **[408, Fig 4, This shows after the I/O request is performed the process is done and there is no writing of the initialization progress into the memory disks]**.

Randall discloses the use of an initialization progress table **[Fig 5]**.

With regard to **claim 23**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** comprises steps of:

detecting, when the RAID **[100, Fig 1; Column 5, Lines 19 - 21]** receives an I/O, whether one of the initialization regions that are associated with the I/O has not been started with the regional initialization **[406 and 407, Fig 4]**;

performing the regional initialization on said initialization region that is associated with the I/O first if said initialization region has not yet started the regional initialization **[408, Fig 4; Column 9, Lines 9 – 18 and 35 – 51, This shows the next step performed from 405 when the initialization for the given region is not completed**

or started is that the initialization is completed for the given region to generate new redundancy information].

With regard to **claim 24**, DeKoning teaches of performing a consecutive consistency initialization **[Fig 3 and Fig 4]** on the initialization regions that have not yet completed the regional initialization **[Fig 3, The figure shows that initialization starts at the beginning of the LUN and ends once it has initialized the whole LUN]**.

4. Claims **2, 4 – 6, 18, and 22** are rejected under 35 U.S.C. 103(a) as being unpatentable over DeKoning et al (Pat 6,467,023) in view of Randall et al. (Pat 6,530,031) as applied to **claim 1** above, and further in view of TechTarget ("Nonvolatile Storage").

With regard to **claim 2**, DeKoning teaches the RAID configuration is stored in a random access memory **[118, Fig 1; Column 5, Lines 64 – 66; Column 6, Lines 1 – 5]**.

However, DeKoning in view of Randall does not specifically disclose the limitation of non-volatile memory to store data.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of TechTarget in DeKoning in view of Randall, because DeKoning is using a random access memory to store the progress of the initialization **[DeKoning, Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** and TechTarget discloses a random access memory that has non-volatile storage

characteristics that would allow the progress of the initialization to be saved even if the power to the system was interrupted **[TechTarget, Page 1]**.

With regard to **claim 4**, DeKoning teaches the steps of:

detecting, when the RAID receives an I/O **["PROCESS I/O REQUEST", Fig 4]**, whether the initialization region(s) that is(are) associated with the I/O is(are) completed with the regional initialization **[405, Fig 4]**;

waiting for completion of the regional initialization if the initialization region(s) is(are) not completed with the regional initialization **407, Fig 4; Column 9, Lines 21 – 23]** and the regional initialization is being performed on the initialization region(s) that is(are) associated with the I/O **[405, Fig 4; Column 9, Lines 9 – 21, If the request is above and below the boundary that indicates that the regional initialization has started but is not complete for the region the I/O is associated with]**;

updating an initialization state change of the initialization region(s) **[Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]**;

writing the updated initialization state change into a memory device **[Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** before an I/O result is returned **[Column 8, Lines 30 – 38; Fig 3; This shows the updated initialization state change information is saved before an I/O result is returned which is in response to an I/O input to restart the initialization]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

However, DeKoning in view of Randall does not specifically disclose the limitation that the memory the updated initialization changed are written to is a non-volatile memory.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of TechTarget in DeKoning in view of Randall, because DeKoning is using a random access memory to store the progress of the initialization **[DeKoning, Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** and TechTarget discloses a random access memory that has non-volatile storage characteristics that would allow the progress of the initialization to be saved even if the power to the system was interrupted **[TechTarget, Page 1]**.

With regard to **claim 5**, DeKoning teaches wherein the I/O accesses the RAID **[100, Fig 1; Column 5, Lines 19 – 21]** after the step of writing the updated initialization into the memory device **[Column 7, Lines 12 – 23, This shows the I/O accesses the RAID before and after a particular update to the progress data since the initialization and I/O accesses are performed in parallel]**.

Randall discloses the use of a table to store initialization information **[500, Fig 5]**.

TechTarget discloses the memory is non-volatile memory **[Page 1]**.

With regard to **claim 6**, DeKoning teaches wherein the I/O accesses the RAID **[100, Fig 1; Column 5, Lines 19 – 21]** before the step of writing the updated initialization into the memory device **[Column 7, Lines 12 – 23, This shows the I/O**

accesses the RAID before and after a particular update to the progress data since the initialization and I/O accesses are performed in parallel].

Randall discloses the use of a table to store initialization information **[500, Fig 5]**.

TechTarget discloses the memory is non-volatile memory **[Page 1]**.

With regard to **claim 18**, DeKoning teaches there are a plurality of versions of the initialization progress **[307, Fig 3, Each save would result in different version of data]** stored in the memory device **[118, Fig 1; Column 5, Lines 64 – 66; Column 6, Lines 1 – 5]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

With regard to **Claim 22**, TechTarget discloses the memory device is a battery backed-up SRAM, a flash RAM **[Page 1]** or a disk drive except a member disk.

5. Claims **8 - 10** are rejected under 35 U.S.C. 103(a) as being unpatentable over DeKoning et al (Pat 6,467,023) in view of Randall et al. (Pat 6,530,031) as applied to **claim 7** above, and further in view of TechTarget ("Nonvolatile Storage").

With regard to **claim 8**, DeKoning teaches the consecutive consistency initialization **[Fig 3 and Fig 4]** comprises steps of:

selecting on of the initialization regions **["PORTION", 300 and 310, Fig 3]** which have not yet been completed with the regional initialization **[310, Fig 3]**;

performing the regional initialization **["PORTION", 300, 302, 304, 306, 307, 308, and 310, Fig 3]** on the initialization region **[308, 310, and 302, Fig 3]** if a regional initialization **["PORTION", 300, 302, 304, 306, 307, 308, and 310, Fig 3]** is not already

being performed on the selected initialization region **[“PORTION”, 300 and 310, Fig 3, The figure shows the process of performing initialization on regions if the initialization has not occurred already until all regions have been initialized]**;

updating an initialization state change of the selected initialization region **[307, Fig 3]**;

writing the updated initialization progress into the memory device **[118, Fig 1; Column 5, Lines 64 - 67; Column 6, Lines 1 - 5]** when the regional initialization is performed at a suitable time, wherein the suitable time is a timing when a predetermined number of initialization regions is completed with the regional initialization **[307, Fig 3; Column 5, Lines 64 – 67; Column 6 – Lines 1 – 5, This shows the updated initialization progress is saved to the random access memory device at a suitable time which is after each portion’s initialization process is complete. The predetermined number of initialization regions is one since after each region is initialized the progress is saved to the memory device]**, when a predetermined percentage of the initialization regions is completed with the regional initialization, or when a predetermined time has elapsed after the initialization progress is stored in a member disk;

repeating aforesaid steps until initialization regions **[“PORTION”, 300 and 310, Fig 3]** have completed with the regional initialization **[308 and 312, Fig 3]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

However, DeKoning in view of Randall does not specifically disclose the limitation that the memory the updated initialization changed are written to is a non-volatile memory.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of TechTarget in DeKoning in view of Randall, because DeKoning is using a random access memory to store the progress of the initialization **[DeKoning, Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** and TechTarget discloses a random access memory that has non-volatile storage characteristics that would allow the progress of the initialization to be saved even if the power to the system was interrupted **[TechTarget, Page 1]**.

With regard to **claim 9**, DeKoning teaches after all the initialization regions have been completed with the regional initialization **[308 and 312, Fig 3]**, step of writing a state which shows that all initialization regions **[“PORTION”, 300 and 310, Fig 3]** are completed with initialization **[312, Fig3]** into the memory device **[Column 5, Lines 64 – 66; Column 6, Lines 1 – 5]**.

TechTarget discloses the use of a non-volatile memory **[Page 1]**.

With regard to **claim 10**, DeKoning teaches the consecutive consistency initialization **[Fig 3 and Fig 4]** comprises the steps of:

performing a regional initialization priority adjustment mechanism to determine whether selecting of the initialization regions **[“PORTION”, 300 and 310, Fig 3]** which

has not yet been completed with the regional initialization **["PORTION", 300 and 310, Fig 3]**;

selecting one of the initialization regions **["PORTION", 300 and 310, Fig 3]** which have not yet been completed with the regional initialization **[310, Fig 3]**;

performing the regional initialization **["PORTION", 300, 302, 304, 306, 307, 308, and 310, Fig 3]** on the selected initialization region **[308, 310, and 302, Fig 3]** if the regional initialization **["PORTION", 300, 302, 304, 306, 307, 308, and 310, Fig 3]** is not being performed on the selected initialization region **["PORTION", 300 and 310, Fig 3, The figure shows the process of performing initialization on regions if the initialization has not occurred already until all regions have been initialized]**;

updating an initialization state change of the selected initialization region **[307, Fig 3]**;

writing the updated initialization progress data into a memory device, **[118, Fig 1; Column 5, Lines 64 - 67; Column 6, Lines 1 - 5]** when the regional initialization is performed at a suitable time, wherein the suitable time is a timing when a predetermined number of initialization regions is completed with the regional initialization **[307, Fig 3; Column 5, Lines 64 – 67; Column 6 – Lines 1 – 5, This shows the updated initialization progress is saved to the random access memory device at a suitable time which is after each portion's initialization process is complete. The predetermined number of initialization regions is one since after each region is initialized the progress is saved to the memory device]**, when a predetermined percentage of the initialization regions is completed with the regional

initialization, or when a predetermined time has elapsed after the initialization progress is stored in a member disk;

repeating aforesaid steps until initialization regions **[“PORTION”, 300 and 310, Fig 3]** have completed with the regional initialization **[308 and 312, Fig 3]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

However, DeKoning in view of Randall does not specifically disclose the limitation that the memory the updated initialization changed are written to is a non-volatile memory.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of TechTarget in DeKoning in view of Randall, because DeKoning is using a random access memory to store the progress of the initialization **[DeKoning, Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** and TechTarget discloses a random access memory that has non-volatile storage characteristics that would allow the progress of the initialization to be saved even if the power to the system was interrupted **[TechTarget, Page 1]**.

6. Claims **15 - 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over DeKoning et al (Pat 6,467,023) in view of Randall et al. (Pat 6,530,031) as applied to **claim 3** above, and further in view of TechTarget (“Nonvolatile Storage”).

With regard to **claim 15**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** comprises dividing a data space of member disks **[108, Fig 1]** into a plurality

of initialization regions **[“PORTION”, 300 and 310, Fig 3]** and performing the regional initialization on the initialization regions **[“PORTION”, 300, 302, 304, 306, 307, 308, and 310, Fig 3]**, and after the I/O that induces the regional initialization completes access to a data space of the RAID, the initialization progress data is written into a memory device, and then an I/O result is returned **[Figs 3 – 4; Column 5, Lines 54 – 67; Column 6, Lines 1 – 5; Column 9, Lines 9 – 17, This shows the process of an I/O request being receive that is for access to a section of memory not initialized section of memory which then forces the desired memory location to perform initialization and due to the initialization the progress is saved in a random access memory and then the return of the I/O is if the entire LUN is initialized or not]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

However, DeKoning in view of Randall does not specifically disclose the limitation of non-volatile memory to store data.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of TechTarget in DeKoning in view of Randall, because DeKoning is using a random access memory to store the progress of the initialization **[DeKoning, Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** and TechTarget discloses a random access memory that has non-volatile storage characteristics that would allow the progress of the initialization to be saved even if the power to the system was interrupted **[TechTarget, Page 1]**.

With regard to **claim 16**, DeKoning teaches the consistency initialization **[Fig 3 and Fig 4]** comprises dividing a data space of member disks **[108, Fig 1]** into a plurality of initialization regions **[“PORTION”, 300 and 310, Fig 3]** and performing the regional initialization on the initialization regions **[“PORTION”, 300, 302, 304, 306, 307, 308, and 310, Fig 3]** and after the initialization progress is first written in to a memory device **[118, Fig 1; Column 5, Lines 64 – 66; Column 6, Lines 1 – 5]**, an I/O **[“INITIALIZE LUN”, Fig 3]** accesses the data space of the RAID **[Fig 3; 404 and 406, Fig 4, This shows an I/O request that is granted access to the data space of the RAID when the data space requested by the I/O request is above the boundary line which means the initialization for the data space has already happened wish then also means the progress data for that data space has been saved to memory]**.

Randall discloses the use of a table to store initialization state data **[500, Fig 5]**.

However, DeKoning in view of Randall does not specifically disclose the limitation of non-volatile memory to store data.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of TechTarget in DeKoning in view of Randall, because DeKoning is using a random access memory to store the progress of the initialization **[DeKoning, Column 5, Lines 64 – 67; Column 6, Lines 1 – 5]** and TechTarget discloses a random access memory that has non-volatile storage characteristics that would allow the progress of the initialization to be saved even if the power to the system was interrupted **[TechTarget, Page 1]**.

7. Claim **17** is rejected under 35 U.S.C. 103(a) as being unpatentable over DeKoning et al (Pat 6,467,023) in view of Randall et al. (Pat 6,530,031) in view of TechTarget (“Nonvolatile Storage”) as applied to **claim 2** above, and further in view of Humlicek et al. (Pat 5,822,782).

With regard to **claim 17**, DeKoning teaches the RAID configuration is stored in a random access memory **[118, Fig 1; Column 5, Lines 64 – 66; Column 6, Lines 1 – 5]**.

TechTarget discloses the use of non-volatile memory to store data **[Page 1]**.

However, DeKoning in view of Randall in view of TechTarget does not specifically disclose the limitation that the non-volatile memory is a member disk.

Humlicek discloses a RAID system that stores configuration information on the disk drives **[Column 6, Lines 52 - 62]**.

It would have been obvious to someone of ordinary skill in the art at the time of the invention to use the teachings of Humlicek in DeKoning in view of Randall in view of TechTarget, because there is a limited number of non-volatile type memories and both a memory disk from Humlicek and the memory disclosed by TechTarget provide examples of non-volatile member and it would be a design choose as to which type of non-volatile memory to store the data and both would be expected to provide the same property of retaining the memory once the main power source is removed from the system.

Response to Amendment/Arguments

8. Applicant's arguments filed 12/06/2010 with regard to claims 1 – 24 have been fully considered but they are not persuasive.

The Applicant argues on pages 8 - 15 with regard to claim 1 that Randall teaches a timing table utilized for self-test of each hardware during the boot time of a computer and calculating the elapsed time of each task **[Page 11, Lines 3 – 5]** in contrast to the initialization progress table of the present invention which includes a plurality of fields for storing initialization states of each of a plurality of initialization regions of a RAID so as to indicate which initialization regions have been initialized by a regional initialization and which initialization regions have not yet been initialized **[Page 11, Lines 11 - 15]** and the Applicant argues a specific layout of an initialization progress table that has multiple rows with four columns where one column is the initialization region and the other 3 columns contain a bit indicating a status of the initialization with respect to the initialization region associated in the same row **[Page 12 - 15]**. After careful consideration of the Applicant's arguments the Examiner respectfully disagrees with the Applicant.

Randall discloses a table that is used to maintain status information about initialization progress in computer hardware. Randall is not used to teach what the initialization process is. DeKoning is used to teach the specific method of initializing a RAID memory structure so the data is consistent as shown in the rejection of claim 1 above. There is no limitation in claim 1 as to what exactly the states of the initialization are and how many bits they are. Claim 1 does not specifically define the initialization

table or consistency initialization as indicated by the Applicant. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., The table being designed as argued and shown in figures 1 – 2B on pages 13 - 15) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Claim 1 requires, with respect to the initialization progress table, a table structure that contains at least 2 fields for at least two different area or regions and the fields contain information on the state of an initialization process. There is no specific limitation as to how many fields there are exactly or how the states are specifically stored in the table. DeKoning teaches initializing multiple regions in a RAID memory system. Randall teaches a table structure that is used to monitor an initialization process that keeps track of the time it takes to perform a desired initialization and the table can be used to determine if there is a problem with the hardware the initialization is being applied to. The claim does not specifically state how the state of the initialization is maintained in the table other then states of the initializations. Randall monitors initialization progress by monitoring the time of the initialization. The time lets a user know if an initialization is taking longer then expected letting the user know there is a possible issue with the hardware that is being initialized. This allows a user of the DeKoning RAID system know if there is a hard drive in the RAID that is possibly failing

and to replace it before the hard drive completely fails or that a hard drive has completely failed and needs to be replaced.

The Applicant argues on page 16 that claims 2 – 24 are allowable for being dependent on claim 1 which the Applicant has argued above as being allowable. After careful consideration of the Applicant's arguments the Examiner respectfully disagrees.

The Examiner has responded to the arguments regarding claim 1 above showing how the prior art reads on the claim limitations in claim 1. Therefore, the rejections of claims 2 – 24 are maintained at this time.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Lubbers et al. (Pat 6,161,192) discloses a RAID system **[Fig 1]** and initializing the RAID involves setting data regions in the RAID to all the same value **[148, Fig 4; Column 5, Line 65 through Column 6, Line 15, This shows in step 146 in the initialization process the data region's FE Bit are all set to "0"]**.

Schultz et al. (Pat 5,909,691) discloses determining if all disk in a set of disk are consistent with each other **[Fig 10; Column 35, Line 58 through Column 36, Line 34]**.

Direction Of Future Correspondence

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTOPHER D. BIRKHIMER whose telephone number is (571)270-1178. The examiner can normally be reached on M-H 7:00 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Christopher D Birkhimer
Examiner
Art Unit 2186

/Matt Kim/
Supervisory Patent Examiner, Art Unit 2186

/Christopher D Birkhimer/
Examiner, Art Unit 2186